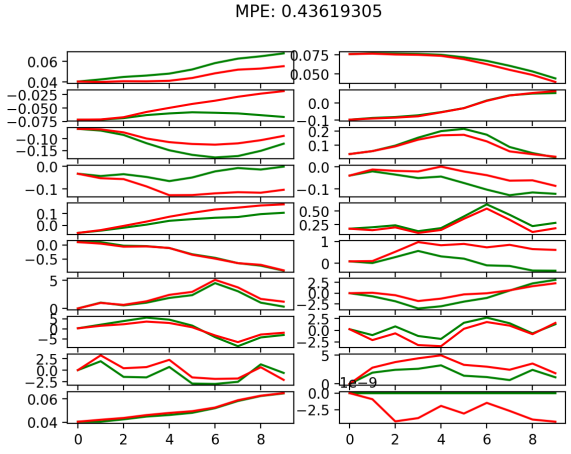
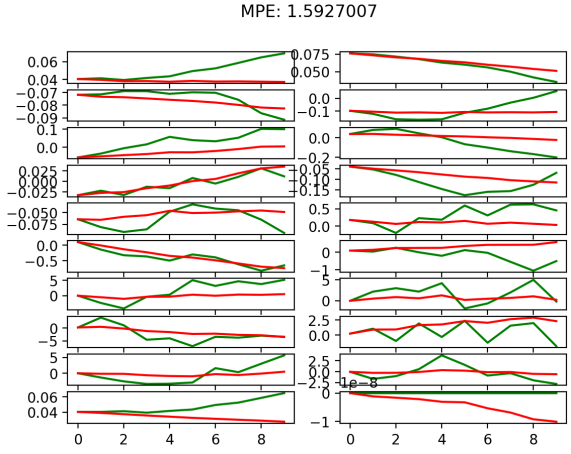
Homework 2

Model Based Reinforcement Learning

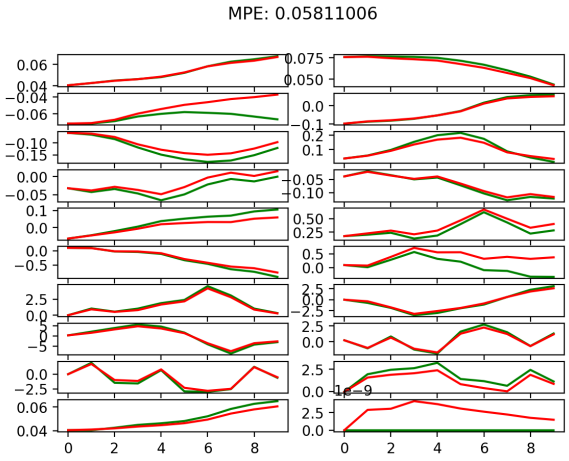
Problem 1



First model



Second model



Third model

Figure 1: MPE (Model prediction error) across different models on the half-cheetah task. The first model was training with a single layer of 32 neurons and a training steps number of 500. The second one had 2 layers of 250 neurons and a training steps number of 5. The last model had 2 layers of 250 neurons with a training steps number of 500.

By looking at Figure 1 we can infer that the best model is the third one as its MPE is the lowest one ($\simeq 0.06$). This can be explained by a larger capacity and training steps number than the other 2 models, which can lead to learn better representation and therefore make better predictions.

Clearly, the second model is the worst one with an MPE $\simeq 1.6$. It can be the resulting from the agent's training update per iteration set to 5, which limits the learning process even if the agent's neural network has a better capacity than the first model.

rem: One could raise the fact that a too low number of training steps with a high capacity neural network seems leading to more errors than a neural network with a small capacity and high training steps number to learn the dynamics of the environment.

Problem 2

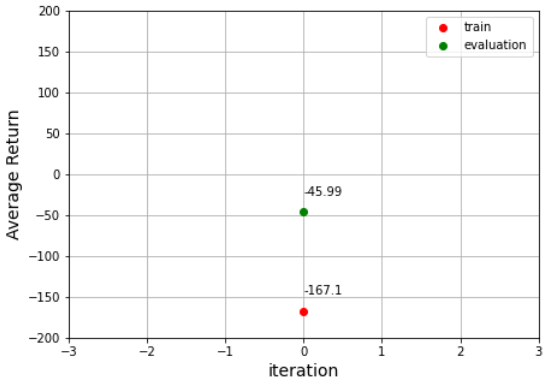
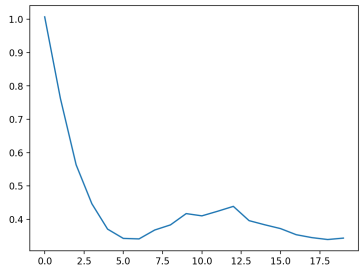


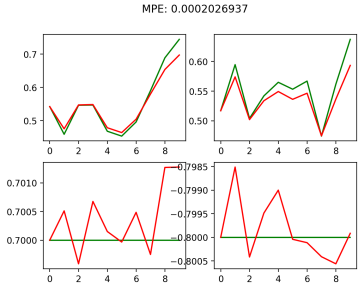
Figure 2: Training and evaluation average return of the first iteration from MPC policy using random shooting on obstacles task. The number of training steps per iteration was set to 20, the initial batch size to 5000, the batch size to 1000, and the action sequence predictive horizon to 10.

Problem 3

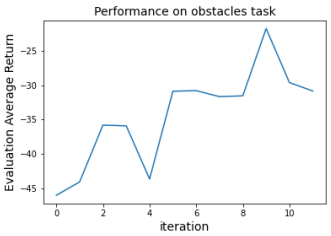
3.1 Obstacles environment



Training loss as function of iteration.



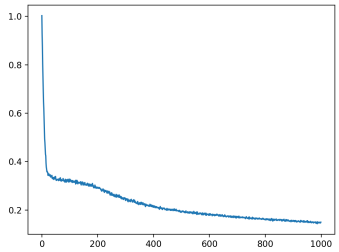
Dynamics model prediction performance. Each plot corresponds to a different state element.



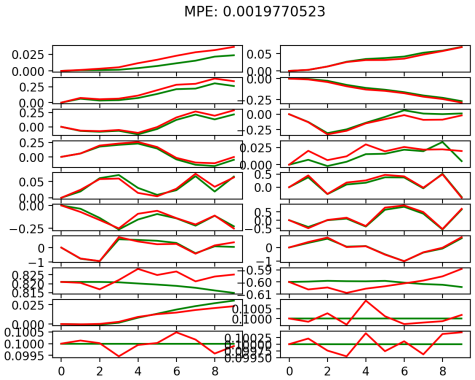
MPE evaluation average return as function of iteration.

Figure 3: Different performance metrics of the agent on the obstacles task. The number of training steps per iteration was set to 20, the initial batch size to 5000, the batch size to 1000, the action sequence predictive horizon to 10, and the number of iteration was 12.

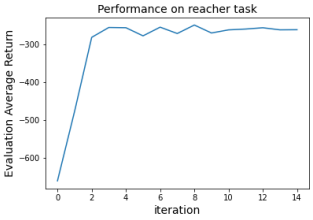
3.2 Reacher environment



Training loss as function of iteration.



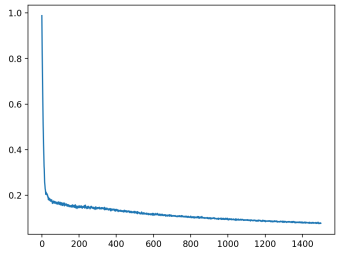
Dynamics model prediction performance. Each plot corresponds to a different state element.



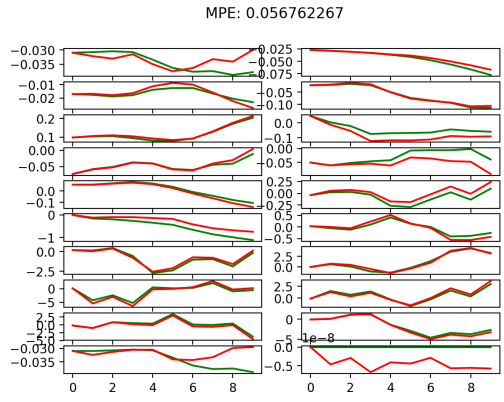
MPE evaluation average return as function of iteration.

Figure 4: Different performance metrics of the agent on the reacher task. The number of training steps per iteration was set to 1000, the initial batch size to 5000, the batch size to 5000, the action sequence predictive horizon to 10, and the number of iteration was 15.

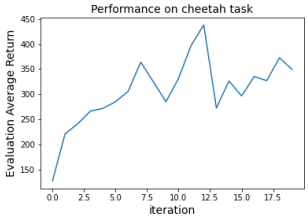
3.3 Half-Cheetah environment



Training loss as function of iteration.



Dynamics model prediction performance. Each plot corresponds to a different state element.



MPE evaluation average return as function of iteration.

Figure 5: Different performance metrics of the agent on the half-cheetah task. The number of training steps per iteration was set to 1500, the initial batch size to 5000, the batch size to 5000, the action sequence predictive horizon to 15, and the number of iteration was 20.

Problem 4

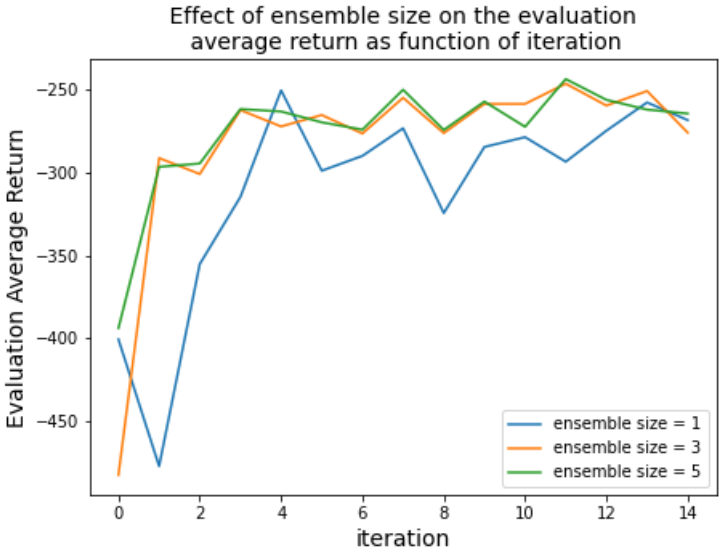


Figure 6: Effect of the ensemble size on the evaluation average return on reacher task. When there is only one model making a prediction, this seems to have a notifiable variance on the average return of the evaluation, and the average return is not as good as for an ensemble size of 3 or 5. It is also interesting to note that there is no significant difference between aggregating the predictions of 3 models or 5. The optimal choice here would be to use an ensemble size of 3 to reduce the training time of the agent.

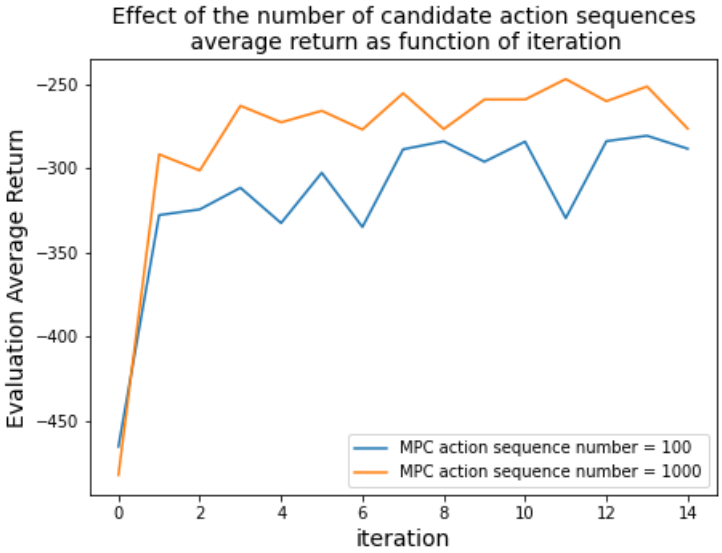


Figure 7: Effect of the number of action sequences sampled to sampling trajectory on the evaluation average rewards as function of iteration on reacher task. Clearly, sampling 1000 action sequences leads to much more rewards ($\pm$ 50 more) than sampling 100. The usefulness of sacrificing computing time for better performance is a trade-off that depends on the task. Indeed, in the case of robotics, if with 250 rewards the agent solves its objective qualitatively valid according to the expert who judges it, then it may be beneficial to sample only 100 action sequences instead of 1000 to save computation time and battery.

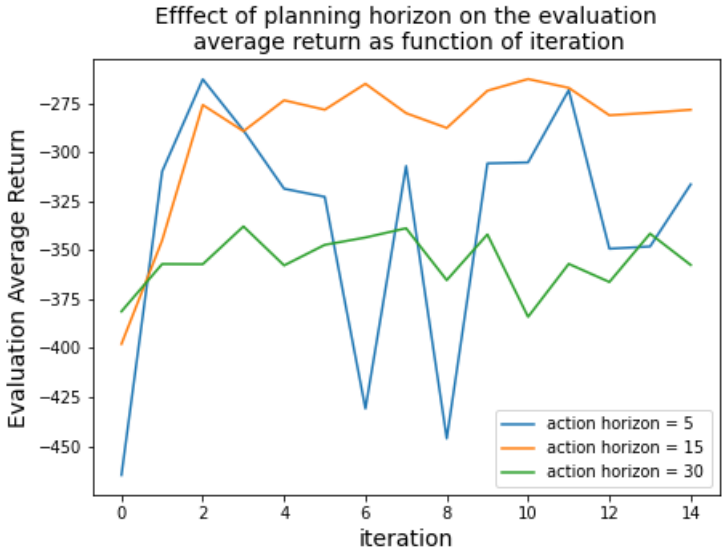


Figure 8: Effect of the planning horizon on the evaluation average return as function of iteration on reacher task.We can see that planning without looking enough into the future leads to bad choices or at least increase the variance in the predictions, as shown by the agent trained with a horizon of 5. On the contrary, planning by looking too far into the future reduces the quality of predictions due to the curse of horizon, as shown by the agent trained with a horizon of 30. It is necessary to find the middle ground as shown by the agent trained with a horizon of 15 which arrives at much better results than its counterparts.

Problem 5

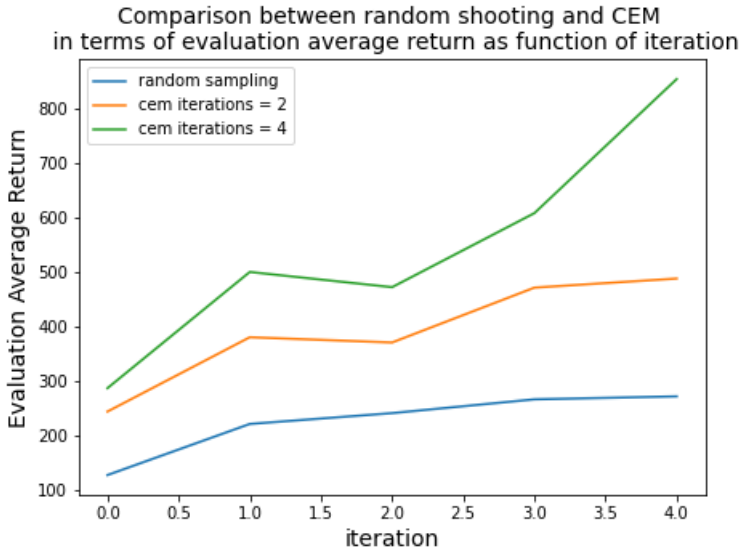


Figure 9: Comparison between the evaluation average return resulting from different number of iteration for CEM as function of agent's training iteration on half-cheetah task. It can be seen that as the number of iterations of CEM sampling increases, the more it converges to a policy that produces better results. Doubling the number of iterations of CEM from 2 to 4 almost doubles the evaluation average return (500 for CEM iteration = 2 and 800 for CEM iteration = 4) by the 4th iteration of the agent's training.That makes sens as with 4 iterations of CEM we can refine 2 times our elites action sequences candidates mean and standard deviation which leads to be closer the optimal distributions of action sequences to solve the task.